Tight Post-Quantum Multi-Key Security of the FX Construction

Rentaro Shiba[1,2] and Tetsu Iwata[1]

[1] Nagoya University, Nagoya, Japan,
shiba.rentaro.k7@s.mail.nagoya-u.ac.jp, tetsu.iwata@nagoya-u.jp
[2] Mitsubishi Electric Corporation, Kamakura, Japan,
Shiba.Rentaro@dc.MitsubishiElectric.co.jp

Abstract. The FX construction (FX) is one of the simplest key-length extension schemes for block ciphers, introduced by Kilian and Rogaway as a generalization of DESX. By extending the key length without modifying the underlying block cipher, FX can improve security against adversaries that perform quantum computation but are restricted to classical queries to the keyed construction. Recently, Shiba and Iwata showed in their ToSC 2026 paper that FX with an n-bit block and a κ-bit internal key is secure up to approximately $2^{(n+\kappa-\log_2 \mu)/3}$ total classical and quantum oracle queries in the quantum ideal cipher model, where $\mu \leq 2^\kappa$ is the number of independently keyed instances. In this paper, we improve this bound to approximately $2^{(n+\kappa)/3}$ total oracle queries in the same model against fully adaptive adversaries, for every $\mu \geq 1$ and $1 \leq \kappa \leq 2n$. The known attack based on the offline Simon's algorithm can target a single keyed instance and matches our bound in the exponent. Thus, the exponent of the total query complexity required to break FX is independent of the number of keyed instances in this parameter regime.

Keywords: symmetric-key cryptography · FX construction · provable security · post-quantum security · multi-key security · multi-user security

1 Introduction

Post-quantum security of symmetric-key cryptosystems. Recent advances in quantum computing have raised concerns about the security of existing cryptosystems. While some public-key cryptosystems are known to be severely threatened by quantum algorithms for the discrete logarithm problem and integer factorization [25], the implications for symmetric-key cryptosystems are less widely recognized. Nevertheless, quantum algorithms such as Grover's algorithm [12] can significantly reduce security margins by providing a quadratic speedup for the exhaustive key search.

To assess post-quantum security, two adversary models are commonly studied. Following the terminology of Kaplan et al. [16], these are referred to as the

Q1 and Q2 models. In the Q1 model, the adversary performs quantum computation but makes only classical queries to the secretly keyed construction. The Q2 model additionally allows quantum superposition queries to the construction.

Idealized security analyses model internal primitives of the target constructions as random oracles, random permutations, or ideal ciphers and grant the adversary quantum access to them. These settings are called the quantum random oracle model (QROM) [8], quantum random permutation model (QRPM) [14], and quantum ideal cipher model (QICM) [14], respectively.

In practical applications, many cryptographic systems use the same symmetric-key construction under multiple secret keys. For example, a TLS 1.3 server handling many connections protected by AES-GCM uses multiple traffic keys derived by the TLS key schedule [22, Sections 7.1 and 7.3]. Another example is volume encryption with BitLocker, which can use XTS-AES to protect different storage volumes under volume-specific keys [20,19]. Shiba and Iwata [23] also discuss EMV-compliant payment systems[1] as a practical motivation for the multi-key setting. These applications motivate security analyses involving multiple keyed instances, with independent uniform keys serving as an abstraction under suitable assumptions on key generation and derivation.

The post-quantum security of these applications must also be considered. Indeed, NIST has standardized ML-KEM [21], and hybrid key-establishment mechanisms combining classical key exchange with ML-KEM are being deployed in TLS [11]. Even when key establishment and derivation remain secure against quantum adversaries, the symmetric-key constructions protecting data under the resulting keys must still withstand quantum attacks. An adversary may obtain data from many keyed instances through classical interactions while evaluating the symmetric-key algorithm under candidate keys on a quantum computer. It is therefore important to study post-quantum security in the multi-key setting and determine how the security guarantee depends on the number of keys in use.

Recently, post-quantum security in the multi-key setting has been studied in several quantum idealized models. Shiba and Iwata [23] analyzed tweakable Even-Mansour in the QRPM and the FX construction in the QICM. They also studied tweakable key-alternating Feistel ciphers in the QROM [24]. Hosoyamada [13] analyzed GCM in the QICM. These analyses consider adversaries with classical access to multiple independently keyed instances and quantum access to their shared public primitive.

The FX construction. Key-length extension mitigates the security loss caused by quantum key search. The FX construction (FX) is one of the simplest key-length extension schemes for block ciphers. It extends the key length by XORing an independent whitening key into the input and output of a block cipher, without modifying the block cipher itself. FX was introduced by Kilian and Rogaway [17] as a generalization of DESX[2].

[1] `https://www.emvco.com/`

[2] DESX was originally proposed by Ron Rivest, but his original proposal was unpublished.

Let $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be an ideal block cipher, and let $K = (k, k') \in \{0,1\}^{\kappa+n}$. The encryption $\mathsf{FX}[E] : \{0,1\}^{\kappa+n} \times \{0,1\}^n \to \{0,1\}^n$ (Figure 1) is defined as

$$\mathsf{FX}[E](K, x) \coloneqq E_k(x \oplus k') \oplus k'.$$

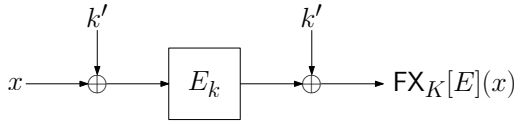

Fig. 1. FX

Here, k is the internal ideal-cipher key and k' is the whitening key. We may write $\mathsf{FX}[E](K, \cdot) = \mathsf{FX}_K[E](\cdot)$ in the rest of the paper. Related constructions based on key whitening are used in concrete block ciphers such as PRINCE [10], PRIDE [5], and MANTIS [7].

Security of the FX construction. Kilian and Rogaway [17] proved that FX is secure up to approximately $2^{(\kappa+n)/2}$ classical queries in the ideal cipher model.

In the Q2 model, Leander and May [18] proposed the Grover-Meets-Simon attack, which recovers the secret key of FX using $\widetilde{O}(2^{\kappa/2})$[3] quantum queries. This has the same exponent as Grover's exhaustive key search against the underlying block cipher, so the whitening key provides no improvement in this exponent. In contrast, when access to the keyed construction is restricted to classical queries, i.e., in the Q1 model, key whitening can improve security against quantum adversaries. The following results establish the security of FX against Q1 adversaries in the QICM.

In this setting, Jaeger et al. [15] established security up to approximately $2^{(\kappa+n)/3}$ total oracle queries for $1 \le \kappa \le 2n$, under the restriction that the classical construction queries are fixed before the execution begins. Subsequently, Alagic et al. [3] and, independently, Shiba and Iwata [23] established the same single-key security level against fully adaptive adversaries. On the cryptanalytic side, Bonnetain et al. [9] proposed the offline Simon's algorithm and applied it to a key-recovery attack against FX in the Q1 model. Their attack uses $\widetilde{O}(2^{(\kappa+n)/3})$ total oracle queries, so the known single-key query lower bound is tight in the exponent.

Shiba and Iwata [23] also extended their analysis to the multi-key setting. For $\mu \le 2^\kappa$ independently keyed instances, they established security up to approximately $2^{(\kappa+n-\log_2 \mu)/3}$ total oracle queries in the QICM. Here, the query count includes classical queries across all keyed instances and quantum queries

[3] We use $\widetilde{O}$ to suppress factors polynomial in n.

to the shared ideal cipher. Compared with the single-key result, this query lower bound is smaller by a factor of $\mu^{1/3}$. However, the tightness of their multi-key bound was left open.

This raises the question of whether the dependence on the number of keys reflects an actual weakness of FX or a limitation of the existing proof. We revisit their analysis and ask whether this dependence can be removed while retaining the single-key security level against adaptive quantum adversaries.

1.1 Contribution

We prove that, for every $\mu \geq 1$ and $1 \leq \kappa \leq 2n$, FX is secure up to approximately $2^{(\kappa+n)/3}$ total classical and quantum oracle queries in the multi-key setting by refining the previous security proof of Shiba and Iwata [23]. This security bound is tight in the exponent. Our result holds in the QICM against fully adaptive adversaries with classical forward and inverse access to the independently keyed construction oracles and quantum forward and inverse access to the ideal cipher.

In the same setting, Shiba and Iwata [23] established security up to approximately $2^{(\kappa+n-\log_2 \mu)/3}$ total oracle queries for $\mu \leq 2^{\kappa}$. Their proof adapts the hybrid framework introduced by Alagic et al. [1] to the multi-key setting. The hybrids replace classical construction responses one at a time with responses from independent random permutations, while reprogramming the ideal primitive to be consistent with the resulting classical transcript. As in Basak et al. [6], their proof separates the probability of a bad event in the ideal experiment from the distinguishing advantages between adjacent hybrids. The adjacent hybrids are then compared using arbitrary reprogramming [1, Lemma 3] and the ideal-cipher resampling lemma of Alagic et al. [2, Lemma 2].

For the bad event, their analysis first accounts for collisions among the initially sampled internal keys and then bounds the probability of a corresponding ideal-cipher input or output collision. In the hybrid comparisons, the probability estimates also account for a cipher key matching any of the μ internal keys. These estimates contain additional factors of μ. Their resulting multi-key query lower bound is smaller than the single-key bound by a factor of $\mu^{1/3}$.

We remove this loss by refining the bad-event probability estimate and the hybrid comparisons using arbitrary reprogramming and ideal-cipher resampling. For the bad event, we evaluate the joint probability of an internal ideal-cipher key collision and a corresponding ideal-cipher input or output collision for each pair of classical queries. For arbitrary reprogramming, we similarly retain the internal ideal-cipher key condition when evaluating the contribution of each classical query. These estimates remove the additional factors of μ from the bad-event probability bound and the reprogramming probability bound.

For ideal-cipher resampling, we use the generalized ideal-cipher resampling lemma of Alagic et al. [3, Lemma 2] in place of the lemma in [2, Lemma 2]. The generalized lemma allows the resampling distribution to depend on the preceding interactions. We construct its distinguisher so that the sampled key is assigned as the internal ideal-cipher key associated with the oracle selected by the next classical query. This assignment preserves the distribution of the secret

Table 1. Query lower bounds for FX against fully adaptive adversaries in the QICM, for $1 \leq \kappa \leq 2n$. Queries include classical construction queries and quantum ideal-cipher queries.

Reference	Number of keys	Query lower bound
[3,23]	$\mu = 1$	$\Omega(2^{(\kappa+n)/3})$
[23]	$1 \leq \mu \leq 2^{\kappa}$	$\Omega(2^{(\kappa+n-\log_2 \mu)/3})$
Theorem 1	$\mu \geq 1$	$\Omega(2^{(\kappa+n)/3})$

keys because the preceding interactions use independent random permutations and the public ideal cipher. Possible collisions involving preceding queries to other classical oracles are bounded separately. This construction removes the additional dependence on μ from the resampling comparison.

Together, these refinements recover the single-key query lower bound for an arbitrary number of independently keyed instances sharing the same ideal cipher. The resulting advantage bound depends on the total numbers of classical and quantum queries and the upper bound on classical queries to each construction oracle, with no explicit dependence on μ.

As we mentioned above, our query lower bound is tight in the exponent. The offline Simon's algorithm-based attack of Bonnetain et al. [9] requires $O(2^{(\kappa+n)/3})$ classical queries and $\widetilde{O}(2^{(\kappa+n)/3})$ quantum queries to the ideal cipher. For every $\mu \geq 1$, this attack applies in the multi-key setting by targeting a single keyed construction oracle. Its total query complexity therefore matches our lower bound in the exponent.

Table 1 summarizes the known post-quantum query lower bounds for FX and our result.

1.2 Organization

Section 2 introduces our notation, adversary model, and security notion, and recalls the arbitrary reprogramming and ideal-cipher resampling lemmas. Section 3 presents our security bound and its proof, followed by a discussion of the resulting query lower bound and its tightness. Section 4 concludes the paper.

2 Preliminaries

2.1 Notation

For integers $a \leq b$, let $[a, b] \coloneqq \{a, a+1, \ldots, b\}$, and write $[b] \coloneqq [1, b]$. We set $[0] \coloneqq \emptyset$. For a finite set $\mathcal{X}$ (resp. a distribution D), we write $x \leftarrow_{\$} \mathcal{X}$ (resp. $x \leftarrow_{\$} D$) to denote that x is sampled uniformly at random from $\mathcal{X}$ (resp. according to D). We let $\mathsf{Perm}(n)$ denote the set of all permutations over $\{0,1\}^n$. We let $\mathsf{Cipher}(\kappa, n)$ denote the set of all ciphers over key space $\{0,1\}^{\kappa}$ and block space $\{0,1\}^n$. For $a, b \in \{0,1\}^n$, let $\mathsf{swap}_{a,b} : \{0,1\}^n \to \{0,1\}^n$ denote the

permutation that exchanges a and b and fixes all other points. If $a = b$, then $\mathsf{swap}_{a,b}$ is the identity permutation.

2.2 Adversary Model and Security Notion

Fix $\mu \geq 1$, and let $\mathcal{A}$ be a quantum adversary. In the real-world experiment, sample $E \leftarrow_{\$} \mathsf{Cipher}(\kappa, n)$ and $K_1 = (k_1, k_1'), \ldots, K_\mu = (k_\mu, k_\mu') \leftarrow_{\$} \{0,1\}^\kappa \times \{0,1\}^n$. The adversary has classical forward and inverse access to $\mathsf{FX}_{K_1}[E], \ldots, \mathsf{FX}_{K_\mu}[E]$ and quantum forward and inverse access to E. In the ideal-world experiment, replace the construction oracles by independent $\pi_1, \ldots, \pi_\mu \leftarrow_{\$} \mathsf{Perm}(n)$, independent also of E. We define the post-quantum multi-key strong pseudorandom permutation advantage of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{pq\text{-}mk\text{-}sprp}}_{\mathsf{FX}}(\mathcal{A}) \coloneqq \left| \Pr\left[\mathcal{A}^{\{\mathsf{FX}_{K_u}[E]\}_{u\in[\mu]}, |E\rangle} = 1\right] - \Pr\left[\mathcal{A}^{\{\pi_u\}_{u\in[\mu]}, |E\rangle} = 1\right] \right|.$$

Here, $|E\rangle$ denotes quantum oracle access to both E and E^{-1} through the unitary operations

$$|k\rangle|x\rangle|z\rangle \mapsto |k\rangle|x\rangle|z \oplus E_k(x)\rangle, |k\rangle|y\rangle|z\rangle \mapsto |k\rangle|y\rangle|z \oplus E_k^{-1}(y)\rangle,$$

respectively. We consider fully adaptive adversaries. The adversary may choose which keyed oracle to query next, as well as the input, query direction, and timing of the query, based on all preceding classical and quantum interactions.

The adversary makes at most q_C classical queries in total, at most d classical queries to each keyed oracle, and at most q_Q quantum queries to the ideal cipher. We assume that the adversary never makes a classical query whose answer is already determined by its preceding queries to the same classical oracle.

2.3 Arbitrary Reprogramming

The arbitrary reprogramming (ARP) lemma was introduced in [1]. It considers a distinguisher that fixes a base oracle and a randomized algorithm that outputs a set of reprogramming points. The distinguisher then receives quantum access either to the base oracle or the reprogrammed oracle.

ARP experiment. Let $\mathcal{D}$ be a distinguisher that runs the following ARP game.

Phase 1: $\mathcal{D}$ specifies a function $F_0 \coloneqq F : \{0,1\}^n \to \{0,1\}^m$ and a randomized algorithm $\mathcal{B}$ which outputs a set $B \subset \{0,1\}^n \times \{0,1\}^m$ where each $x \in \{0,1\}^n$ is the first element of at most one tuple in B. Let $B_1 \coloneqq \{x \mid \exists y : (x,y) \in B\}$.

Phase 2: $\mathcal{B}$ is run to obtain B. Let $F_1 \coloneqq F^{(B)}$, where $F^{(B)}$ is a locally redefined version of F, defined as

$$F^{(B)}(x) \coloneqq \begin{cases} y & \text{if } (x,y) \in B, \\ F(x) & \text{otherwise.} \end{cases}$$

The challenger samples a uniform bit $b \in \{0,1\}$ and $\mathcal{D}$ is given quantum access to F_b.

Phase 3: $\mathcal{D}$ loses access to F_b, and receives the randomness r used to invoke $\mathcal{B}$. Then $\mathcal{D}$ outputs b', a guess of b.

Lemma 1 (ARP lemma [1, Lemma 3]). *Define the reprogramming probability by* $\epsilon \coloneqq \max_{x\in\{0,1\}^n}\{\Pr_{B\leftarrow_\$\mathcal{B}}[x \in B_1]\}$. *Then, for any* $\mathcal{D}$ *making at most* q *queries in expectation during Phase 2 when its oracle is* F_0, *it holds that*

$$\mathsf{Adv}^{\mathsf{arp}}(\mathcal{D}) \coloneqq |\mathsf{Pr}[\mathcal{D}=1 \mid b=1] - \mathsf{Pr}[\mathcal{D}=1 \mid b=0]| \le 2q\sqrt{\epsilon}. \tag{1}$$

The proof of Lemma 1 is given in [1, Section 4.1].

2.4 Ideal-Cipher Resampling

The ideal-cipher resampling (IC-RS) lemma of Alagic et al. [3] considers a distinguisher that first makes quantum queries to an ideal cipher and then specifies a distribution over a cipher key and two input points. A triple is sampled from this distribution, and the images of the two points under the corresponding keyed permutation are either swapped or left unchanged. The distinguisher then receives quantum access to the resulting cipher.

IC-RS experiment. Let $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ be a quantum distinguisher in the following experiment.

Phase 1: Sample $E \leftarrow_\$ \mathsf{Cipher}(\kappa, n)$ and provide $\mathcal{D}_0$ with quantum oracle access to E and E^{-1}. It outputs a description of a distribution D on $\{0,1\}^\kappa \times \{0,1\}^n \times \{0,1\}^n$.

Phase 2: Sample $(k, s_0, s_1) \leftarrow_\$ D$. Set $E^{(0)} \coloneqq E$, and define

$$E^{(1)}_{k^*}(x) \coloneqq \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k, \\ (E_{k^*} \circ \mathsf{swap}_{s_0,s_1})(x) & \text{if } k^* = k. \end{cases}$$

Sample a uniform bit $b \leftarrow_\$ \{0,1\}$. Give $\mathcal{D}_1$ the triple (k, s_0, s_1) and quantum oracle access to $E^{(b)}$ and $(E^{(b)})^{-1}$. It then outputs b', a guess of b.

Lemma 2 (IC-RS lemma [3, Lemma 2]). *For a distribution* D *output in Phase 1, define the resampling probability by*

$$\epsilon \coloneqq \max_{(k,s_0,s_1)\in\{0,1\}^\kappa\times\{0,1\}^n\times\{0,1\}^n} D(k, s_0, s_1).$$

Then, if $\mathcal{D}_0$ *makes at most* q *queries in Phase 1, it holds that*

$$\mathsf{Adv}^{\mathsf{ic\text{-}rs}}(\mathcal{D}) \coloneqq |\Pr[\mathcal{D}=1 \mid b=1] - \Pr[\mathcal{D}=1 \mid b=0]| \le 4\sqrt{2^n q\epsilon}.$$

The proof of Lemma 2 is given in Appendix A.1 of [4], the full version of [3].

3 Tight Post-Quantum Multi-Key Security of FX

In this section, we prove the following theorem.

Theorem 1 (Post-quantum multi-key security of FX). *For every $\mu \geq 1$, let $E \leftarrow_{\$} \mathsf{Cipher}(\kappa, n)$, $\pi_1, \ldots, \pi_\mu \leftarrow_{\$} \mathsf{Perm}(n)$, and $K_1, \ldots, K_\mu \leftarrow_{\$} \{0,1\}^{\kappa+n}$ be sampled independently. For any adversary $\mathcal{A}$ making at most q_C classical queries in total to either $\mathsf{FX}_{K_1}[E], \ldots, \mathsf{FX}_{K_\mu}[E]$ or $\pi_1, \ldots, \pi_\mu$, with at most d ($d \in [1, \min\{q_C, 2^n\}]$) queries to each classical oracle, and at most q_Q quantum queries to E, the following bound holds:*

$$\mathsf{Adv}^{\mathsf{pq\text{-}mk\text{-}sprp}}_{\mathsf{FX}}(\mathcal{A}) \leq 8q_C\sqrt{\frac{q_Q}{2^\kappa(2^n - d + 1)}} + 2q_Q\sqrt{\frac{2q_C}{2^{\kappa+n}}} + \frac{2q_C(q_C-1)}{2^{\kappa+n}}. \quad (2)$$

Our proof follows the hybrid argument of Alagic et al. [1], as adapted to the multi-key setting by Shiba and Iwata [23, Section 4.2]. We also incorporate techniques from the analysis of Alagic et al. [3]. Section 3.1 gives an overview of the proof and highlights our improvements over the previous multi-key analysis. Section 3.2 presents the full proof.

3.1 Proof Overview and Improvements

Following Alagic et al. [1], for an execution with $q_C \geq 1$ classical queries, we divide the quantum computation and quantum queries into stages $0, \ldots, q_C$. For $1 \leq j < q_C$, stage j begins after the j-th classical response and ends when the $(j+1)$-st classical query is submitted. Stage 0 covers the computation and quantum queries before the first classical query is submitted, and stage q_C covers those after the last classical response and before the final output.

We introduce $q_C + 1$ hybrid experiments $\mathsf{H}_0, \ldots, \mathsf{H}_{q_C}$, where H_0 is the real-world experiment and H_{q_C} is the ideal-world experiment. In H_j, the first j classical queries are answered using independent random permutations, while the remaining classical queries are answered using FX. The ideal cipher is re-programmed to be consistent with the classical transcript obtained from the random permutations. The formal definitions of these experiments are given in Section 3.2.

We bound the distinguishing advantage by comparing adjacent hybrids H_j and H_{j+1} for $0 \leq j < q_C$. Following the previous post-quantum security proof of FX by Shiba and Iwata [23], we first separate the probability of the bad event in the ideal experiment H_{q_C} from the distinguishing advantages between adjacent hybrids[4]. After bounding this probability, we compare each pair through an intermediate experiment H'_j. The comparison between H'_j and H_{j+1} uses the ARP lemma (Lemma 1) to change the quantum oracle during stage $j+1$. The comparison between H_j and H'_j uses the IC-RS lemma (Lemma 2) to change the next classical response together with the cipher used for all subsequent queries.

[4] A similar separation of the bad-event probability in the ideal world from the hybrid comparisons is also used in [6,24].

Although we follow this overall approach, we refine the following three parts of the proof: the analysis of the bad event in the ideal world, the hybrid comparison using ARP, and the hybrid comparison using IC-RS.

Bad event in the ideal world. In the previous proof of Shiba and Iwata [23], the probability of the bad event is bounded by $O(\mu^2 q_C^2/2^{\kappa+n})$. Their estimate considers collisions among the initially sampled internal keys, followed by collisions among the corresponding inputs or outputs of the ideal cipher. The factor of μ^2 in their bound arises from counting pairs of initially sampled internal keys. We instead evaluate, for each pair of classical queries, the probability that their corresponding internal keys coincide and that their corresponding inputs or outputs of the ideal cipher also coincide. We then sum these probabilities over all pairs of classical queries.

Under the non-redundancy assumption of classical queries in Section 2.2, two queries to the same classical oracle cannot cause the bad event. For each pair of queries to different classical oracles, the probability that their internal keys coincide and that their corresponding inputs or outputs of the ideal cipher also coincide is at most $2/2^{\kappa+n}$, since the secret keys remain independent and uniform after fixing the ideal transcript. There are at most $\binom{q_C}{2}$ pairs of classical queries, so we obtain $O({q_C}^2/2^{\kappa+n})$ as the upper bound on the probability.

Hybrid comparison using ARP. In the previous proof of Shiba and Iwata [23], the reprogramming probability in the comparison between H'_j and H_{j+1} is bounded by $O(\mu(j+1)/2^{\kappa+n})$. The factor of μ arises from accounting for a fixed ideal-cipher key matching any of the μ internal keys.

We instead evaluate the contribution of each classical query, keeping its internal ideal-cipher key condition together with the condition on the corresponding reprogramming point. For a fixed direction, cipher key, and block, each classical query contributes at most two possible reprogramming points under its associated internal key. Since the internal key and whitening key remain independent and uniform after fixing the preceding ideal transcript, the contribution of each classical query is at most $2/2^{\kappa+n}$. Summing over the $j+1$ classical queries gives an upper bound of $O((j+1)/2^{\kappa+n})$ on the reprogramming probability, removing the additional factor of μ from the previous estimate.

Hybrid comparison using IC-RS. Shiba and Iwata [23] use an earlier ideal-cipher resampling lemma [2, Lemma 2], in which the cipher key and the two points are sampled independently and uniformly. In their application of this lemma, the resampling probability is $O(\mu/2^{\kappa+n})$. The factor of μ arises from accounting for the key used for resampling matching any of the μ internal keys.

We instead assign the cipher key supplied by the IC-RS experiment as the internal key associated with the oracle selected by the next classical query. This preserves the distribution of the secret keys, since the preceding interactions do not use them.

We use the IC-RS lemma of Alagic et al. [3], which allows the cipher key and the two points to be sampled from a distribution that depends on the preceding interactions. The cipher key and the first point remain independent and uniform, while the second point is sampled uniformly after excluding the points corresponding to preceding queries to the selected oracle. The new lemma permits this exclusion. At most $d-1$ points are excluded, so the maximum probability of sampling any particular cipher key and pair of points is $O\left(\frac{1}{2^{\kappa+n}(2^n-d+1)}\right)$. Applying the IC-RS lemma gives a distinguisher advantage of $O(\sqrt{q_Q/(2^\kappa(2^n-d+1))})$, with no explicit dependence on μ. Possible collisions with preceding queries to other classical oracles require both an internal-key match and an input or output match. Their total probability after j classical queries is $O(j/2^{\kappa+n})$.

3.2 Proof of Theorem 1

Proof. Let $\tau_j \coloneqq ((u_1, x_1, y_1, \delta_1), \ldots, (u_j, x_j, y_j, \delta_j))$ be the classical transcript for the first j classical queries. For each $i \in [j]$, $u_i \in [\mu]$ is the index of the classical oracle to which the adversary made its i-th classical query, x_i is the plaintext, y_i is the ciphertext, and $\delta_i \in \{0,1\}$ is the direction of the i-th classical query. The query is answered using π_{u_i} or the FX oracle instantiated with K_{u_i}, as prescribed by the experiment. If $\delta_i = 0$, the i-th classical query is a forward query, with input x_i and response y_i. If $\delta_i = 1$, the i-th classical query is an inverse query, with input y_i and response x_i. Following [3,23], set $E^0 \coloneqq E$ and recursively define the modified cipher by

$$(E^j)_k(x) \coloneqq \begin{cases} (\mathsf{swap}_{(E^{j-1})_k(x_j \oplus k'_{u_j}), y_j \oplus k'_{u_j}} \circ (E^{j-1})_k)(x) & \text{if } k = k_{u_j}, \\ (E^{j-1})_k(x) & \text{if } k \neq k_{u_j}. \end{cases}$$

Define col_j as the event that:

$$\begin{aligned} \mathsf{col}_j : &\text{ there exist distinct } i, i' \in [j] \text{ with } k_{u_i} = k_{u_{i'}} \\ &\text{such that } x_i \oplus k'_{u_i} = x_{i'} \oplus k'_{u_{i'}} \quad \text{or} \quad y_i \oplus k'_{u_i} = y_{i'} \oplus k'_{u_{i'}}. \end{aligned}$$

If col_j does not occur, then E^j is consistent with the transcript τ_j, i.e.,

$$(E^j)_{k_{u_i}}(x_i \oplus k'_{u_i}) = y_i \oplus k'_{u_i} \qquad \text{for every } i \in [j].$$

We now define a sequence of hybrid experiments H_j for all $j \in [0, q_C]$.

Experiment H_j. Sample $\pi_1, \ldots, \pi_\mu \leftarrow_\$ \mathsf{Perm}(n)$, $E \leftarrow_\$ \mathsf{Cipher}(\kappa, n)$, and $K_1 = (k_1, k'_1), \ldots, K_\mu = (k_\mu, k'_\mu) \leftarrow_\$ \{0,1\}^{\kappa+n}$. Then:

1. Run $\mathcal{A}$, answering its classical queries using $\pi_1, \ldots, \pi_\mu$ and its quantum queries using E. Stop immediately before making its $(j+1)$-st classical query.
2. Resume $\mathcal{A}$. For the remainder of $\mathcal{A}$'s execution,
   - If col_j occurs:
     - $\mathcal{A}$'s classical queries are answered with an error symbol $\perp$.

- $\mathcal{A}$'s quantum queries are answered using E.

– Else:
- $\mathcal{A}$'s classical queries are answered using $\mathsf{FX}_{K_1}[E^j], \ldots, \mathsf{FX}_{K_\mu}[E^j]$.
- $\mathcal{A}$'s quantum queries are answered using E^j.

Under the assumption that col_j does not occur, we can concisely describe experiment H_j as providing oracle responses to $\mathcal{A}$ according to the following sequence:

$$\underbrace{E, \pi_{u_1}, E, \cdots, \pi_{u_j}, E}_{j \text{ classical queries}}, \underbrace{\mathsf{FX}_{K_{u_{j+1}}}\left[E^j\right], E^j, \cdots, \mathsf{FX}_{K_{u_{q_C}}}\left[E^j\right], E^j}_{q_C - j \text{ classical queries}}.$$

In these sequences, each occurrence of π_{u_i} or $\mathsf{FX}_{K_{u_i}}[E^j]$ for each $i \in [q_C]$ represents a single classical query, while each occurrence of E or E^j represents the quantum queries made during the corresponding stage. H_0 corresponds to the experiment in the real world, and H_{q_C} corresponds to the experiment in the ideal world. Therefore, we have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{pq\text{-}mk\text{-}sprp}}_{\mathsf{FX}}(\mathcal{A}) &= \left|\Pr\left[\mathcal{A}^{\{\mathsf{FX}_{K_u}[E]\}_{u\in[\mu]},|E\rangle} = 1\right] - \Pr\left[\mathcal{A}^{\{\pi_u\}_{u\in[\mu]},|E\rangle} = 1\right]\right| \\
&= |\Pr[\mathcal{A}(\mathsf{H}_0) = 1] - \Pr[\mathcal{A}(\mathsf{H}_{q_C}) = 1]| \\
&= |\Pr[\mathcal{A}(\mathsf{H}_0) = 1] - \Pr[\mathcal{A}(\mathsf{H}_{q_C}) = 1 \wedge \neg\mathsf{col}_{q_C}] \\
&\qquad\qquad - \Pr[\mathcal{A}(\mathsf{H}_{q_C}) = 1 \wedge \mathsf{col}_{q_C}]| \\
&\leq |\Pr[\mathcal{A}(\mathsf{H}_0) = 1] - \Pr[\mathcal{A}(\mathsf{H}_{q_C}) = 1 \wedge \neg\mathsf{col}_{q_C}]| + \Pr[\mathsf{col}_{q_C}] \\
&\leq \left(\sum_{j=0}^{q_C-1} |\Pr[\mathcal{A}(\mathsf{H}_j) = 1 \wedge \neg\mathsf{col}_j] - \Pr[\mathcal{A}(\mathsf{H}_{j+1}) = 1 \wedge \neg\mathsf{col}_{j+1}]|\right) \\
&\qquad\qquad + \Pr[\mathsf{col}_{q_C}]
\end{aligned} \tag{3}$$

We first bound the probability of col_{q_C}. This corresponds to a bad event for the classical transcript in the ideal world. For the probability of col_{q_C}, we have the following lemma.

Lemma 3.

$$\Pr[\mathsf{col}_{q_C}] \leq \frac{q_C(q_C-1)}{2^{\kappa+n}}.$$

Proof. We condition on a fixed classical transcript τ_{q_C} generated in the ideal-world experiment H_{q_C}. Since this transcript is independent of $K_1, \ldots, K_\mu$, these keys remain independent and uniform. By the non-redundancy assumption, no pair of classical queries to the same oracle can cause col_{q_C}. Therefore, for any distinct $i, i' \in [q_C]$ with $u_i \neq u_{i'}$, we have

$$\Pr\left[k_{u_i} = k_{u_{i'}} \wedge x_i \oplus k'_{u_i} = x_{i'} \oplus k'_{u_{i'}}\right] = 2^{-(\kappa+n)}.$$

The same holds with $x_i, x_{i'}$ replaced by $y_i, y_{i'}$. Let $q_{C,u}$ be the number of τ_{q_C} records from the u-th classical oracle (π_u). Then,

$$\Pr[\mathsf{col}_{q_C}] \leq \frac{2}{2^{\kappa+n}} \sum_{\substack{u,u' \in [\mu] \\ u<u'}} q_{C,u} q_{C,u'} \leq \frac{q_C(q_C-1)}{2^{\kappa+n}}.$$

The bound holds for every ideal transcript, and hence also after averaging over τ_{q_C}. □

We next compare H_j and H_{j+1}. To make the comparison easy, we introduce the intermediate experiment H'_j for $j \in [0, q_C - 1]$.

Experiment H'_j. Sample $\pi_1, \ldots, \pi_\mu \leftarrow_\$ \mathsf{Perm}(n)$, $E \leftarrow_\$ \mathsf{Cipher}(\kappa, n)$, and $K_1 = (k_1, k'_1), \ldots, K_\mu = (k_\mu, k'_\mu) \leftarrow_\$ \{0,1\}^{\kappa+n}$. Then:

1. Run $\mathcal{A}$, answering its classical queries using $\pi_1, \ldots, \pi_\mu$ and its quantum queries using E. Stop immediately after receiving the response to $\mathcal{A}$'s $(j+1)$-st classical query.
2. Resume $\mathcal{A}$. For the remainder of $\mathcal{A}$'s execution,
   - If col_{j+1} occurs:
     - $\mathcal{A}$'s classical queries are answered with an error symbol $\perp$.
     - $\mathcal{A}$'s quantum queries are answered using E.
   - Else:
     - $\mathcal{A}$'s classical queries are answered using $\mathsf{FX}_{K_1}[E^{j+1}], \ldots, \mathsf{FX}_{K_\mu}[E^{j+1}]$.
     - $\mathcal{A}$'s quantum queries are answered using E^{j+1}.

Under the assumption that col_{j+1} does not occur, we can concisely describe experiment H'_j as providing oracle responses to $\mathcal{A}$ according to the following sequence:

$$\underbrace{E, \pi_{u_1}, E, \cdots, \pi_{u_j}, E}_{j \text{ classical queries}}, \pi_{u_{j+1}}, E^{j+1}, \underbrace{\mathsf{FX}_{K_{u_{j+2}}}\left[E^{j+1}\right], E^{j+1}, \cdots, \mathsf{FX}_{K_{u_{q_C}}}\left[E^{j+1}\right], E^{j+1}}_{q_C - j - 1 \text{ classical queries}}.$$

By introducing this experiment, we can efficiently compare H_j and H_{j+1} as

$$\begin{aligned}
&|\Pr[\mathcal{A}(\mathsf{H}_j) = 1 \wedge \neg\mathsf{col}_j] - \Pr[\mathcal{A}(\mathsf{H}_{j+1}) = 1 \wedge \neg\mathsf{col}_{j+1}]| \\
&\leq \left|\Pr[\mathcal{A}(\mathsf{H}_j) = 1 \wedge \neg\mathsf{col}_j] - \Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg\mathsf{col}_{j+1}\right]\right| \\
&\qquad + \left|\Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg\mathsf{col}_{j+1}\right] - \Pr[\mathcal{A}(\mathsf{H}_{j+1}) = 1 \wedge \neg\mathsf{col}_{j+1}]\right|.
\end{aligned}$$

We complete the proof by bounding the two differences displayed above.

We first compare H_{j+1} and H'_j. For these experiments, we have the following.

Lemma 4. *For all $j \in [0, q_C - 1]$,*

$$\left|\Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg\mathsf{col}_{j+1}\right] - \Pr[\mathcal{A}(\mathsf{H}_{j+1}) = 1 \wedge \neg\mathsf{col}_{j+1}]\right| \leq 2q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{\kappa+n}}},$$

where $q_{Q,j+1}$ denotes the expected number of quantum queries made during stage $j+1$ in the ideal experiment H_{q_C}. Since the adversary makes at most q_Q quantum queries in total, we have $\sum_{j=0}^{q_C-1} q_{Q,j+1} \leq q_Q$.

Proof. Under the assumption that col_{j+1} does not occur, we can concisely describe experiments H_{j+1} and H'_j as providing oracle responses to $\mathcal{A}$ according to the following oracle sequences:

$$\mathsf{H}_{j+1} : \underbrace{E, \pi_{u_1}, E, \ldots, \pi_{u_j}, E}_{j \text{ classical queries}}, \pi_{u_{j+1}}, E, \quad \underbrace{\mathsf{FX}_{K_{u_{j+2}}}\left[E^{j+1}\right], E^{j+1}, \ldots, \mathsf{FX}_{K_{u_{q_C}}}\left[E^{j+1}\right], E^{j+1}}_{q_C - j - 1 \text{ classical queries}}$$

$$\mathsf{H}'_j : \underbrace{E, \pi_{u_1}, E, \ldots, \pi_{u_j}, E}_{j \text{ classical queries}}, \pi_{u_{j+1}}, E^{j+1}, \underbrace{\mathsf{FX}_{K_{u_{j+2}}}\left[E^{j+1}\right], E^{j+1}, \ldots, \mathsf{FX}_{K_{u_{q_C}}}\left[E^{j+1}\right], E^{j+1}}_{q_C - j - 1 \text{ classical queries}}.$$

If col_{j+1} occurs, these experiments give identical oracle responses. Otherwise, these experiments H'_j and H_{j+1} differ only in the cipher used to answer quantum queries at the $(j+1)$-st stage. The initially sampled E is used in the experiment H_{j+1}, while the modified version E^{j+1} is used in the experiment H'_j. Therefore, the distinguishing advantage of $\mathcal{A}$ is bounded by the distinguishing advantage of the ARP experiment with E and its modified version E^{j+1}. Thus, we bound this difference using Lemma 1. From $\mathcal{A}$, we construct an ARP distinguisher $\mathcal{D}$ which proceeds as follows.

Phase 1: $\mathcal{D}$ samples $\pi_1, \ldots, \pi_\mu \leftarrow_\$ \mathsf{Perm}(n)$, $E \leftarrow_\$ \mathsf{Cipher}(\kappa, n)$. It then runs $\mathcal{A}$, answering its classical queries using $\pi_1, \ldots, \pi_\mu$ and quantum queries using E. It stops immediately after receiving a response to $\mathcal{A}$'s $(j+1)$-st classical query. $\mathcal{D}$ defines $F_0(a, k, x) = F(a, k, x) := E^a_k(x)$ for $(a, k, x) \in \{-1, 1\} \times \{0,1\}^\kappa \times \{0,1\}^n$. It also specifies a randomized algorithm $\mathcal{B}$ that samples $K_1 = (k_1, k'_1), \ldots, K_\mu = (k_\mu, k'_\mu) \leftarrow_\$ \{0,1\}^{\kappa+n}$ and outputs the set B of input and output pairs to be reprogrammed so that $F_1(a, k, x) = F^{(B)}(a, k, x) = (E^{j+1})^a_k(x)$. Finally, $\mathcal{D}$ outputs $(F, \mathcal{B})$.

Phase 2: $\mathcal{B}$ is run to obtain B. A uniform bit $b \in \{0, 1\}$ is chosen and $\mathcal{D}$ is given quantum access to F_b. $\mathcal{D}$ resumes $\mathcal{A}$, answering its quantum queries during the $(j+1)$-st stage using F_b. Phase 2 ends at the end of the $(j+1)$-st stage.

Phase 3: $\mathcal{D}$ loses access to F_b and receives randomness used by $\mathcal{B}$ to generate keys. If col_{j+1} occurs, it outputs 0. Otherwise, it resumes $\mathcal{A}$, answering its classical queries using $\mathsf{FX}_{K_1}[E^{j+1}], \ldots, \mathsf{FX}_{K_\mu}[E^{j+1}]$ and quantum queries using E^{j+1}, and outputs whatever $\mathcal{A}$ outputs.

If col_{j+1} occurs, $\mathcal{D}$ outputs 0 for either challenge bit. On $\neg\mathsf{col}_{j+1}$, the challenge bit 1 gives H'_j and 0 gives H_{j+1}. We now apply Lemma 1. The reprogramming

set B is generated from τ_{j+1} and $K_1, \ldots, K_\mu$ so that B_1 is given by

$$B_1 \coloneqq \left\{ \begin{array}{l} (1, k_{u_i}, x_i \oplus k'_{u_i}), \\ (1, k_{u_i}, E^{-1}_{k_{u_i}}(y_i \oplus k'_{u_i})), \\ (-1, k_{u_i}, y_i \oplus k'_{u_i}), \\ (-1, k_{u_i}, E_{k_{u_i}}(x_i \oplus k'_{u_i})) \end{array} \right\}_{i \in [j+1]}$$

If $(1, k, x) \notin B_1$, then $x \neq x_i \oplus k'_{u_i}$ and $E_k(x) \neq y_i \oplus k'_{u_i}$ for every $i \in [j+1]$ with $k_{u_i} = k$. By induction on i, we have $(E^i)_k(x) = E_k(x)$ for every $i \in [0, j+1]$. In particular, $(E^{j+1})_k(x) = E_k(x)$. The same argument applies to inverse queries.

The probability that any particular input (a, k, x) is reprogrammed is at most the probability that it lies in the set B_1. Fix E, τ_{j+1}, and an input (a, k, x). For each $i \in [j+1]$, the condition $k_{u_i} = k$ holds with probability $2^{-\kappa}$. Conditioned on this equality, each of the two points in direction a equals x for exactly one value of the uniform whitening key k'_{u_i}. Therefore, each classical query contributes at most $2/2^{\kappa+n}$. Taking a union bound, the reprogramming probability ϵ can be bounded by

$$\epsilon = \max_{(a,k,x) \in \{-1,1\} \times \{0,1\}^\kappa \times \{0,1\}^n} \Pr[(a, k, x) \in B_1] \leq \frac{2(j+1)}{2^{\kappa+n}}.$$

Recall that $q_{Q,j+1}$ is the expected number of quantum queries made during the $(j+1)$-st stage. When $b = 0$, the quantum queries in Phase 2 are answered using E. Thus, the execution through the end of the $(j+1)$-st stage is identical to that in the ideal experiment H_{q_C}. The expected number of quantum queries made by $\mathcal{D}$ in Phase 2 when its oracle is F_0 is therefore $q_{Q,j+1}$. Applying Lemma 1, we obtain

$$\begin{aligned} &\left|\Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg \mathsf{col}_{j+1}\right] - \Pr[\mathcal{A}(\mathsf{H}_{j+1}) = 1 \wedge \neg \mathsf{col}_{j+1}]\right| \\ &= \mathsf{Adv}^{\mathsf{arp}}(\mathcal{D}) = |\Pr[\mathcal{D} = 1 \mid b = 1] \ - \Pr[\mathcal{D} = 1 \mid b = 0]| \leq 2q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{\kappa+n}}}. \end{aligned}$$

□

We next compare H_j and H'_j. For these experiments, we have the following.

Lemma 5. *For all $j \in [0, q_C - 1]$,*

$$\begin{aligned} &\left|\Pr[\mathcal{A}(\mathsf{H}_j) = 1 \wedge \neg \mathsf{col}_j] - \Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg \mathsf{col}_{j+1}\right]\right| \\ &\qquad \leq 8\sqrt{\frac{q_Q}{2^\kappa(2^n - d + 1)}} + \frac{2j}{2^{\kappa+n}}. \end{aligned}$$

Proof. For each $j \in [0, q_C - 1]$, to compare H_j and H'_j efficiently, we introduce three additional intermediate experiments H^1_j, H^2_j and H^3_j.

Experiment H^1_j. Sample $\pi_1, \ldots, \pi_\mu \leftarrow_\$ \mathsf{Perm}(n)$, $E \leftarrow_\$ \mathsf{Cipher}(\kappa, n)$, and $K_1 = (k_1, k'_1), \ldots, K_\mu = (k_\mu, k'_\mu) \leftarrow_\$ \{0,1\}^{\kappa+n}$. Then:

1. Run $\mathcal{A}$, answering its classical queries using $\pi_1, \ldots, \pi_\mu$ and its quantum queries using E. Stop immediately after submitting the $(j+1)$-st classical query without receiving its answer.
2. If $\delta_{j+1} = 0$, set $s_0 := x_{j+1} \oplus k'_{u_{j+1}}$ and sample $s_1 \leftarrow_\$ \{0,1\}^n \setminus \{x_i \oplus k'_{u_{j+1}} : i \in [j], u_i = u_{j+1}\}$. If $\delta_{j+1} = 1$, set $s_0 := y_{j+1} \oplus k'_{u_{j+1}}$ and sample $s_1 \leftarrow_\$ \{0,1\}^n \setminus \{y_i \oplus k'_{u_{j+1}} : i \in [j], u_i = u_{j+1}\}$. Define

$$E^{(1)}_k(x) := \begin{cases} (E_k \circ \mathsf{swap}_{s_0,s_1})(x) & \text{if } k = k_{u_{j+1}} \wedge \delta_{j+1} = 0, \\ (\mathsf{swap}_{s_0,s_1} \circ E_k)(x) & \text{if } k = k_{u_{j+1}} \wedge \delta_{j+1} = 1, \\ E_k(x) & \text{if } k \neq k_{u_{j+1}}. \end{cases}$$

   Here, $(E^{(1)})^j$ is obtained by applying the definition of E^j with $E^{(1)}$ in place of E, using τ_j and $K_1, \ldots, K_\mu$.
3. Resume $\mathcal{A}$. For the remainder of $\mathcal{A}$'s execution,
   - If col_j occurs:
     - $\mathcal{A}$'s classical queries are answered with an error symbol $\perp$.
     - $\mathcal{A}$'s quantum queries are answered using E.
   - Else:
     - $\mathcal{A}$'s classical queries are answered using $\mathsf{FX}_{K_1}[(E^{(1)})^j], \ldots, \mathsf{FX}_{K_\mu}[(E^{(1)})^j]$.
     - $\mathcal{A}$'s quantum queries are answered using $(E^{(1)})^j$.

Experiment H^2_j. Step 1 and Step 2 are identical to those of H^1_j. Replace Step 3 with the following:

- If col_j occurs:
  - $\mathcal{A}$'s classical queries are answered with an error symbol $\perp$.
  - $\mathcal{A}$'s quantum queries are answered using E.
- Else:
  (a) If there exists $i \in [j]$ with $u_i \neq u_{j+1}$ and $k_{u_i} = k_{u_{j+1}}$ such that either $\delta_{j+1} = 0$ and $x_i \oplus k'_{u_i} \in \{s_0, s_1\}$, or $\delta_{j+1} = 1$ and $y_i \oplus k'_{u_i} \in \{s_0, s_1\}$, terminate the experiment and output 0.
  (b) Else: Resume $\mathcal{A}$. For the remainder of $\mathcal{A}$'s execution, classical queries are answered using $\mathsf{FX}_{K_1}[(E^{(1)})^j], \ldots, \mathsf{FX}_{K_\mu}[(E^{(1)})^j]$, and quantum queries are answered using $(E^{(1)})^j$.

Experiment H^3_j. This experiment is identical to H^2_j except that the procedure (b) is replaced by the following:

(b') Else: Answer $\mathcal{A}$'s $(j+1)$-st classical query using $\mathsf{FX}_{K_{u_{j+1}}}[(E^{(1)})^j]$ in the direction specified by δ_{j+1}. Append the resulting $(u_{j+1}, x_{j+1}, y_{j+1}, \delta_{j+1})$ to the classical transcript τ_j to obtain τ_{j+1}. Under the conditions for executing this procedure, the resulting transcript satisfies $\neg\mathsf{col}_{j+1}$. Construct E^{j+1} by reprogramming E using τ_{j+1} and $K_1, \ldots, K_\mu$. For the remainder of $\mathcal{A}$'s

execution, classical queries are answered using $\mathsf{FX}_{K_1}[E^{j+1}], \ldots, \mathsf{FX}_{K_\mu}[E^{j+1}]$ and quantum queries are answered using E^{j+1}.

Assume $\neg\mathsf{col}_j$ in H_j, H_j^1, H_j^2, H_j^3, and $\neg\mathsf{col}_{j+1}$ in H'_j. Conditioned on (b) and (b') being executed in H_j^2 and H_j^3, respectively, these experiments H_j, H_j^1, H_j^2, H_j^3 and H'_j can be concisely described by the following oracle sequences:

$$\mathsf{H}_j : \underbrace{E, \pi_{u_1}, E, \ldots, \pi_{u_j}, E,}_{j \text{ classical queries}} \ \mathsf{FX}_{K_{u_{j+1}}}[E^j], \quad E^j, \quad \underbrace{\mathsf{FX}_{K_{u_{j+2}}}[E^j], E^j, \ldots, \mathsf{FX}_{K_{u_{q_C}}}[E^j], E^j}_{q_C - j - 1 \text{ classical queries}}$$

$$\mathsf{H}_j^1 : \underbrace{E, \pi_{u_1}, E, \ldots, \pi_{u_j}, E,}_{j \text{ classical queries}} \ \mathsf{FX}_{K_{u_{j+1}}}[(E^{(1)})^j], \quad (E^{(1)})^j, \quad \underbrace{\mathsf{FX}_{K_{u_{j+2}}}[(E^{(1)})^j], (E^{(1)})^j, \ldots, \mathsf{FX}_{K_{u_{q_C}}}[(E^{(1)})^j], (E^{(1)})^j}_{q_C - j - 1 \text{ classical queries}}$$

$$\mathsf{H}_j^2 : \underbrace{E, \pi_{u_1}, E, \ldots, \pi_{u_j}, E,}_{j \text{ classical queries}} \ \mathsf{FX}_{K_{u_{j+1}}}[(E^{(1)})^j], \quad (E^{(1)})^j, \quad \underbrace{\mathsf{FX}_{K_{u_{j+2}}}[(E^{(1)})^j], (E^{(1)})^j, \ldots, \mathsf{FX}_{K_{u_{q_C}}}[(E^{(1)})^j], (E^{(1)})^j}_{q_C - j - 1 \text{ classical queries}}$$

$$\mathsf{H}_j^3 : \underbrace{E, \pi_{u_1}, E, \ldots, \pi_{u_j}, E,}_{j \text{ classical queries}} \ \mathsf{FX}_{K_{u_{j+1}}}[(E^{(1)})^j], \quad E^{j+1}, \quad \underbrace{\mathsf{FX}_{K_{u_{j+2}}}[E^{j+1}], E^{j+1}, \ldots, \mathsf{FX}_{K_{u_{q_C}}}[E^{j+1}], E^{j+1}}_{q_C - j - 1 \text{ classical queries}}$$

$$\mathsf{H}'_j : \underbrace{E, \pi_{u_1}, E, \ldots, \pi_{u_j}, E,}_{j \text{ classical queries}} \ \pi_{u_{j+1}}, \quad E^{j+1}, \quad \underbrace{\mathsf{FX}_{K_{u_{j+2}}}[E^{j+1}], E^{j+1}, \ldots, \mathsf{FX}_{K_{u_{q_C}}}[E^{j+1}], E^{j+1}}_{q_C - j - 1 \text{ classical queries}}.$$

We can effectively compare H_j and H'_j by using the intermediate experiments as

$$\begin{aligned}
&\left|\Pr[\mathcal{A}(\mathsf{H}_j) = 1 \wedge \neg\mathsf{col}_j] - \Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg\mathsf{col}_{j+1}\right]\right| \\
&\le \left|\Pr[\mathcal{A}(\mathsf{H}_j) = 1 \wedge \neg\mathsf{col}_j] - \Pr\left[\mathcal{A}(\mathsf{H}_j^1) = 1 \wedge \neg\mathsf{col}_j\right]\right| \\
&+ \left|\Pr\left[\mathcal{A}(\mathsf{H}_j^1) = 1 \wedge \neg\mathsf{col}_j\right] - \Pr\left[\mathcal{A}(\mathsf{H}_j^2) = 1 \wedge \neg\mathsf{col}_j\right]\right| \\
&+ \left|\Pr\left[\mathcal{A}(\mathsf{H}_j^2) = 1 \wedge \neg\mathsf{col}_j\right] - \Pr\left[\mathcal{A}(\mathsf{H}_j^3) = 1 \wedge \neg\mathsf{col}_j\right]\right| \\
&+ \left|\Pr\left[\mathcal{A}(\mathsf{H}_j^3) = 1 \wedge \neg\mathsf{col}_j\right] - \Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg\mathsf{col}_{j+1}\right]\right|.
\end{aligned} \tag{4}$$

Comparison of H_j and H_j^1. If col_j occurs, these experiments give identical oracle responses. Otherwise, they differ only in the cipher used for the $(j+1)$-st classical query and all subsequent oracle queries: H_j uses E^j, whereas H_j^1 uses $(E^{(1)})^j$ obtained by the IC-RS experiment. Thus, we bound this difference using Lemma 2. From $\mathcal{A}$, we construct an IC-RS distinguisher $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ which proceeds as follows.

Phase 1: $\mathcal{D}_0$ receives quantum access to $E \leftarrow_\$ \mathsf{Cipher}(\kappa, n)$ and samples $\pi_1, \ldots, \pi_\mu \leftarrow_\$ \mathsf{Perm}(n)$. It runs $\mathcal{A}$, answering its classical queries using $\pi_1, \ldots, \pi_\mu$ and quantum queries using E. It stops immediately after submitting its $(j+1)$-st classical query without receiving its answer. If the $(j+1)$-st classical query is inverse, i.e., $\delta_{j+1} = 1$, $\mathcal{D}$ terminates and outputs 0. If the $(j+1)$-st classical query is forward, i.e., $\delta_{j+1} = 0$, $\mathcal{D}_0$ outputs a description of a distribution D generated by sampling $k \leftarrow_\$ \{0,1\}^\kappa$ and $s_0 \leftarrow_\$ \{0,1\}^n$, followed by $s_1 \leftarrow_\$ \{0,1\}^n \setminus \{x_i \oplus s_0 \oplus x_{j+1} : i \in [j], u_i = u_{j+1}\}$.

Phase 2: $\mathcal{D}_1$ receives the triple $(k, s_0, s_1) \leftarrow_\$ D$ and quantum access to $E^{(b)}$ for the challenge bit $b \in \{0,1\}$. It sets $k_{u_{j+1}} \coloneqq k$, $k'_{u_{j+1}} \coloneqq s_0 \oplus x_{j+1}$, and $K_{u_{j+1}} \coloneqq (k_{u_{j+1}}, k'_{u_{j+1}})$, and samples K_u for all $u \in [\mu] \backslash \{u_{j+1}\}$ independently and uniformly from $\{0,1\}^{\kappa+n}$.

- If col_j occurs, output 0.
- Else: obtain $(E^{(b)})^j$ by reprogramming $E^{(b)}$ using τ_j and $K_1, \ldots, K_\mu$. It resumes $\mathcal{A}$, answering its classical queries using $\mathsf{FX}_{K_1}[(E^{(b)})^j]$,..., $\mathsf{FX}_{K_\mu}[(E^{(b)})^j]$ and quantum queries using $(E^{(b)})^j$, and outputs whatever $\mathcal{A}$ outputs.

If col_j occurs, $\mathcal{D}$ outputs 0 for either challenge bit. On $\neg\mathsf{col}_j$ and $\delta_{j+1} = 0$, the challenge bit 1 gives H_j^1 and 0 gives H_j. We now apply Lemma 2. The values k and s_0 are sampled independently and uniformly from $\{0,1\}^\kappa$ and $\{0,1\}^n$, respectively. The value s_1 is then sampled uniformly from $\{0,1\}^n$, excluding one point for each preceding classical query to the oracle indexed by u_{j+1}. Recall that d is an upper bound on the number of classical queries to each keyed construction oracle. Since the adversary makes at most d classical queries to each keyed construction oracle, including the pending query, at most $d-1$ points are excluded. Then, for every distribution D output by $\mathcal{D}_0$, the resampling probability can be bounded as

$$\epsilon = \max_{(k,s_0,s_1)\in\{0,1\}^\kappa\times\{0,1\}^n\times\{0,1\}^n} D(k,s_0,s_1) \le \frac{1}{2^{\kappa+n}(2^n-d+1)}.$$

$\mathcal{D}_0$ makes at most q_Q quantum queries. Thus, from Lemma 2, we have

$$\mathsf{Adv}^{\mathsf{ic\text{-}rs}}(\mathcal{D}) = |\Pr[\mathcal{D}=1 \mid b=1] - \Pr[\mathcal{D}=1 \mid b=0]| \le 4\sqrt{\frac{q_Q}{2^\kappa(2^n-d+1)}}.$$

If the $(j+1)$-st classical query is inverse, i.e., $\delta_{j+1} = 1$ in Phase 1, an analogous distinguisher can be constructed by applying the above procedure to E^{-1}. Specifically, x_i and x_{j+1} are replaced by y_i and y_{j+1}, respectively, in the sampling of s_1, and $\mathcal{D}_1$ sets $k'_{u_{j+1}} \coloneqq s_0 \oplus y_{j+1}$. If $\delta_{j+1} = 0$, it terminates and outputs 0 in Phase 1. This gives $\mathsf{swap}_{s_0,s_1} \circ E_{k_{u_{j+1}}}$ as the modified cipher used in H_j^1. The same upper bound on ϵ holds. Let this distinguisher for the case of $\delta_{j+1} = 1$ be $\mathcal{D}'$. Splitting the probability difference according to the direction of the $(j+1)$-st classical query and applying the triangle inequality, we obtain

$$\begin{aligned} &\left|\Pr[\mathcal{A}(\mathsf{H}_j) = 1 \wedge \neg\mathsf{col}_j] - \Pr\left[\mathcal{A}(\mathsf{H}_j^1) = 1 \wedge \neg\mathsf{col}_j\right]\right| \\ &\le \mathsf{Adv}^{\mathsf{ic\text{-}rs}}(\mathcal{D}) + \mathsf{Adv}^{\mathsf{ic\text{-}rs}}(\mathcal{D}') \le 8\sqrt{\frac{q_Q}{2^\kappa(2^n-d+1)}}. \end{aligned} \tag{5}$$

Comparison of H_j^1 and H_j^2. H_j^1 and H_j^2 are identical if Step 3 (b) is executed in H_j^2. These experiments differ only when Step 3 (a) is executed in H_j^2. Thus, the distinguishability is bounded by the probability that Step 3 (a) is executed. The keys K_u for $u \neq u_{j+1}$ are uniform and independent of $K_{u_{j+1}}$ and s_0, s_1. Thus, for each $i \in [j]$ with $u_i \neq u_{j+1}$, the probability that $k_{u_i} = k_{u_{j+1}}$ and $x_i \oplus k'_{u_i} \in \{s_0, s_1\}$ is at most $2/2^{\kappa+n}$. The same bound holds with x_i replaced by y_i for an inverse query. Therefore, a union bound gives

$$\begin{aligned} &\left|\Pr\left[\mathcal{A}(\mathsf{H}_j^1) = 1 \wedge \neg\mathsf{col}_j\right] - \Pr\left[\mathcal{A}(\mathsf{H}_j^2) = 1 \wedge \neg\mathsf{col}_j\right]\right| \\ &\le \Pr\left[\text{Step 3 (a) of } \mathsf{H}_j^2 \text{ is executed}\right] \le \frac{2j}{2^{\kappa+n}}. \end{aligned} \tag{6}$$

Comparison of H_j^2 and H_j^3. H_j^2 and H_j^3 proceed identically if col_j occurs or Step 3 (a) is executed. Thus, it suffices to compare Step 3 (b) in H_j^2 and Step 3 (b') in H_j^3. These procedures give the same response to the $(j+1)$-st classical query. For subsequent queries, H_j^2 uses $(E^{(1)})^j$, whereas H_j^3 uses E^{j+1}. We now show that these two ciphers are identical under the conditions for executing these procedures and thus H_j^2 and H_j^3 are identical.

Assume $\delta_{j+1} = 0$. In this case, $s_0 = x_{j+1} \oplus k'_{u_{j+1}}$ and s_1 differ from every $x_i \oplus k'_{u_i}$ with $i \in [j]$ and $k_{u_i} = k_{u_{j+1}}$. For $u_i = u_{j+1}$, non-redundancy of classical queries ensures $s_0 \neq x_i \oplus k'_{u_i}$ and the definition of the distribution of s_1 ensures $s_1 \neq x_i \oplus k'_{u_i}$. For $u_i \neq u_{j+1}$ with $k_{u_i} = k_{u_{j+1}}$, neither s_0 nor s_1 equals $x_i \oplus k'_{u_i}$, because the condition in Step 3 (a) does not hold. Thus, neither s_0 nor s_1 coincides with any input specified by τ_j for reprogramming the permutation indexed by $k_{u_{j+1}}$, and we have

$$((E^{(1)})^j)_{k_{u_{j+1}}} = (E^j)_{k_{u_{j+1}}} \circ \mathsf{swap}_{x_{j+1} \oplus k'_{u_{j+1}}, s_1}.$$

In both experiments, the response to the $(j+1)$-st classical query is therefore

$$y_{j+1} = ((E^{(1)})^j)_{k_{u_{j+1}}}(s_0) \oplus k'_{u_{j+1}} = (E^j)_{k_{u_{j+1}}}(s_1) \oplus k'_{u_{j+1}}.$$

When Step 3 (b') is executed in H_j^3, for every $i \in [j]$ with $k_{u_i} = k_{u_{j+1}}$, we have $y_{j+1} \oplus k'_{u_{j+1}} \neq y_i \oplus k'_{u_i}$. The inequality follows from $s_1 \neq x_i \oplus k'_{u_i}$, the injectivity of $(E^j)_{k_{u_{j+1}}}$, and the equality $(E^j)_{k_{u_{j+1}}}(x_i \oplus k'_{u_i}) = y_i \oplus k'_{u_i}$, which holds by the definition of E^j. Since $x_{j+1} \oplus k'_{u_{j+1}} \neq x_i \oplus k'_{u_i}$ also holds for every such i, neither equality in the definition of col_{j+1} holds for a pair involving the $(j+1)$-st classical query. Together with $\neg\mathsf{col}_j$, this implies $\neg\mathsf{col}_{j+1}$.

By the definition of E^{j+1}, we have

$$\begin{aligned}(E^{j+1})_{k_{u_{j+1}}} &= \mathsf{swap}_{(E^j)_{k_{u_{j+1}}}(s_0), (E^j)_{k_{u_{j+1}}}(s_1)} \circ (E^j)_{k_{u_{j+1}}} \\ &= (E^j)_{k_{u_{j+1}}} \circ \mathsf{swap}_{x_{j+1} \oplus k'_{u_{j+1}}, s_1} \\ &= ((E^{(1)})^j)_{k_{u_{j+1}}}.\end{aligned}$$

The two ciphers also agree at every other internal key. Therefore, $E^{j+1} = (E^{(1)})^j$.

For $\delta_{j+1} = 1$, we apply the forward-case argument to E^{-1}, exchanging x_i and y_i for every $i \in [j+1]$. This again yields $\neg\mathsf{col}_{j+1}$ and $E^{j+1} = (E^{(1)})^j$.

Therefore, the experiments H_j^2 and H_j^3 are identical and

$$\left|\Pr\left[\mathcal{A}(\mathsf{H}_j^2) = 1 \wedge \neg\mathsf{col}_j\right] - \Pr\left[\mathcal{A}(\mathsf{H}_j^3) = 1 \wedge \neg\mathsf{col}_j\right]\right| = 0. \tag{7}$$

Comparison of H_j^3 and H'_j. If col_j occurs, neither of the events being compared holds. We therefore assume $\neg\mathsf{col}_j$.

First, suppose that $\delta_{j+1} = 0$. In H'_j, the response to the $(j+1)$-st classical query can equivalently be generated by sampling $s_1 \leftarrow_{\$} \{0,1\}^n \setminus \{x_i \oplus k'_{u_{j+1}} : i \in [j],\ u_i = u_{j+1}\}$ and setting $y_{j+1} \coloneqq (E^j)_{k_{u_{j+1}}}(s_1) \oplus k'_{u_{j+1}}$. Since $(E^j)_{k_{u_{j+1}}}$

is bijective and satisfies the input and output relations in τ_j, this response is uniform over $\{0,1\}^n \setminus \{y_i : i \in [j],\ u_i = u_{j+1}\}$. This holds for every fixed E and $K_1, \ldots, K_\mu$ satisfying $\neg\mathsf{col}_j$. Thus, this procedure gives the same response distribution as $y_{j+1} = \pi_{u_{j+1}}(x_{j+1})$ in H'_j. We compare the experiments using this generation of the response in H'_j.

If Step 3 (a) is executed in H^3_j, there exists $i \in [j]$ with $u_i \neq u_{j+1}$ and $k_{u_i} = k_{u_{j+1}}$ such that $x_i \oplus k'_{u_i} \in \{s_0, s_1\}$. We consider the following two cases.

- If $x_i \oplus k'_{u_i} = s_0$, then

$$x_i \oplus k'_{u_i} = s_0 = x_{j+1} \oplus k'_{u_{j+1}}.$$

- If $x_i \oplus k'_{u_i} = s_1$, then

$$\begin{aligned} y_{j+1} \oplus k'_{u_{j+1}} &= (E^j)_{k_{u_{j+1}}}(s_1) \\ &= (E^j)_{k_{u_{j+1}}}(x_i \oplus k'_{u_i}) \\ &= y_i \oplus k'_{u_i}. \end{aligned}$$

Thus, col_{j+1} occurs in H'_j. In these cases, H^3_j outputs 0, while $\neg\mathsf{col}_{j+1}$ is false in H'_j. Otherwise, Step 3 (b') is executed in H^3_j. As shown in the preceding comparison, this procedure gives the response

$$y_{j+1} = (E^j)_{k_{u_{j+1}}}(s_1) \oplus k'_{u_{j+1}}$$

and the resulting transcript satisfies $\neg\mathsf{col}_{j+1}$. Hence, the two experiments give the same response to the $(j+1)$-st classical query. They then construct E^{j+1} from the same transcript and use it for all subsequent classical and quantum queries.

For $\delta_{j+1} = 1$, the same argument applies to E^{-1}, with x_i and y_i exchanged for every $i \in [j+1]$. Therefore,

$$\left|\Pr\left[\mathcal{A}(\mathsf{H}^3_j) = 1 \wedge \neg\mathsf{col}_j\right] - \Pr\left[\mathcal{A}(\mathsf{H}'_j) = 1 \wedge \neg\mathsf{col}_{j+1}\right]\right| = 0. \tag{8}$$

Combining Equation (4) with Equations (5) to (8), we obtain the bound in Lemma 5. □

From Equation (3) and Lemmas 3 to 5, we can bound the advantage of $\mathcal{A}$ as

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{pq\text{-}mk\text{-}sprp}}_{\mathsf{FX}}(\mathcal{A}) \\
&\leq \left(\sum_{j\in[0,q_C-1]} |\Pr[\mathcal{A}(\mathsf{H}_j)=1 \wedge \neg\mathsf{col}_j] - \Pr[\mathcal{A}(\mathsf{H}_{j+1})=1 \wedge \neg\mathsf{col}_{j+1}]| \right) \\
&\qquad\qquad + \Pr[\mathsf{col}_{q_C}] \\
&\leq \sum_{j\in[0,q_C-1]} \left(2q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{\kappa+n}}} + 8\sqrt{\frac{q_Q}{2^\kappa(2^n-d+1)}} + \frac{2j}{2^{\kappa+n}} \right) \\
&\qquad\qquad + \frac{q_C(q_C-1)}{2^{\kappa+n}} \\
&\leq 8q_C\sqrt{\frac{q_Q}{2^\kappa(2^n-d+1)}} + 2q_Q\sqrt{\frac{2q_C}{2^{\kappa+n}}} + \frac{2q_C(q_C-1)}{2^{\kappa+n}}.
\end{aligned}$$

□

3.3 Interpretation of Our Bound

Query lower bound. Theorem 1 implies that achieving a constant distinguishing advantage requires $\Omega(2^{(\kappa+n)/3})$ queries in total for $1 \leq \kappa \leq 2n$. This includes classical queries to all keyed construction oracles or independent random permutations, and quantum queries to the ideal cipher.

We use the condition $\kappa \leq 2n$ to obtain a lower bound on $2^n - d + 1$ which appears in the first term on the right-hand side of Equation (2). Suppose that the total number of queries $q_C + q_Q$ is a sufficiently small constant multiple of $2^{(\kappa+n)/3}$. Since $\kappa \leq 2n$, we have $2^{(\kappa+n)/3} \leq 2^n$, so q_C is also a sufficiently small fraction of 2^n. We may therefore apply Theorem 1 with $d = q_C$, in which case $2^n - d + 1$ is at least a positive constant multiple of 2^n. With this estimate, the upper bound on the advantage in Equation (2) can be made arbitrarily small by choosing the constant factor sufficiently small. This yields the claimed query lower bound.

Previously, Shiba and Iwata [23] established a multi-key query lower bound of $\Omega(2^{(\kappa+n-\log_2\mu)/3})$ for $\mu \leq 2^\kappa$. Our result removes this explicit dependence on μ and holds for every $\mu \geq 1$. Thus, for fixed query budgets q_C, q_Q, and $d\,(\leq q_C)$, our security guarantee incurs no additional loss as the number of keyed instances increases.

The absence of an explicit dependence on μ does not mean that the number of keyed instances is irrelevant in practical deployments. The adversary may collect data from multiple keyed instances, and q_C counts its classical queries across all such instances. For example, if the adversary makes d classical queries to each of μ keyed construction oracles, then $q_C = \mu d$. Increasing the number of instances while maintaining the same number of queries per instance therefore increases the total query count and consequently the upper bound on the advantage. As

deployed systems scale, both the number of keyed instances and the aggregate volume of queries may grow.

This query lower bound does not imply that the running time of an attack is independent of the number of keyed instances.

Tightness. Our bound is tight in the exponent for $1 \leq \kappa \leq 2n$. The offline Simon's algorithm-based attack of Bonnetain et al. [9] recovers the secret key using $O(2^{(\kappa+n)/3})$ classical queries and $\widetilde{O}(2^{(\kappa+n)/3})$ quantum queries to the ideal cipher, in the single-key setting.

In the multi-key setting, an adversary can achieve the same query complexity by targeting a single keyed construction oracle. Together with the lower bound we proved, this establishes the claimed tightness.

4 Conclusion

We established tight post-quantum multi-key security of FX in the QICM against fully adaptive adversaries with classical forward and inverse access to multiple independently keyed instances and quantum forward and inverse access to the ideal cipher. For every $\mu \geq 1$ and $1 \leq \kappa \leq 2n$, FX is secure up to approximately $2^{(\kappa+n)/3}$ total quantum and classical oracle queries. This recovers the single-key security level without restricting the number of keyed instances.

Our proof refines the previous analysis of Shiba and Iwata [23] by improving the estimates for the bad event in the ideal world and the hybrid comparisons using arbitrary reprogramming and ideal-cipher resampling. These refinements remove the explicit dependence on μ from the security bound for fixed total and per-oracle query budgets. The known offline Simon's algorithm-based attack can target a single keyed instance and matches our query lower bound in the exponent. Thus, the additional dependence on μ in the previous bound can be eliminated while retaining tightness in the exponent.

Statement on AI Use

AI-use disclosure: Substantial AI contribution.[5] The core ideas for improving the post-quantum multi-key security bound of FX were developed by the authors. GPT-6.1 Sol was used to assist with checking and refining the technical details of the security proof, including the bad-event and reprogramming probability estimates and the hybrid comparisons using ideal-cipher resampling. It also assisted with checking references and with drafting, organizing, and revising the manuscript. The human authors take full responsibility for the correctness and integrity of this work.

[5] This designation is based on the criteria in Section 2.2 of the IACR Generative AI Policy: `https://iacr.org/policy/AIPolicy.html`.

References

1. Alagic, G., Bai, C., Katz, J., Majenz, C.: Post-quantum security of the Even-Mansour cipher. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT (3). Lecture Notes in Computer Science, vol. 13277, pp. 458–487. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_17
2. Alagic, G., Bai, C., Katz, J., Majenz, C., Struck, P.: Post-quantum security of the (tweakable) FX construction, and applications. Cryptology ePrint Archive, Paper 2022/1097 (2022), https://eprint.iacr.org/archive/2022/1097/20230317:200508.pdf, version 20230317:200508
3. Alagic, G., Bai, C., Majenz, C., Shi, K.: Post-quantum security of block cipher constructions. In: SAC (2026), https://sacworkshop.org/SAC26/preproceedings/SAC2026_1-final18.pdf, to appear
4. Alagic, G., Bai, C., Majenz, C., Shi, K.: Post-quantum security of block cipher constructions. IACR Cryptol. ePrint Arch. **2026**, 209 (2026), https://eprint.iacr.org/2026/209
5. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçin, T.: Block ciphers - focus on the linear layer (feat. PRIDE). In: Garay, J.A., Gennaro, R. (eds.) CRYPTO (1). Lecture Notes in Computer Science, vol. 8616, pp. 57–76. Springer (2014). https://doi.org/10.1007/978-3-662-44371-2_4
6. Basak, J., Bhaumik, R., Chauhan, A.K., Jejurikar, R., Jha, A., Roy, A., Schrottenloher, A., Talnikar, S.: Post-quantum security of key-alternating Feistel ciphers. In: Hanaoka, G., Yang, B. (eds.) ASIACRYPT (1). Lecture Notes in Computer Science, vol. 16245, pp. 446–478. Springer (2025). https://doi.org/10.1007/978-981-95-5018-0_15
7. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO (2). Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5
8. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_3
9. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: The offline Simon's algorithm. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT (1). Lecture Notes in Computer Science, vol. 11921, pp. 552–583. Springer (2019). https://doi.org/10.1007/978-3-030-34578-5_20
10. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Wang, X., Sako, K. (eds.) ASIACRYPT. Lecture Notes in Computer Science, vol. 7658, pp. 208–225. Springer (2012). https://doi.org/10.1007/978-3-642-34961-4_14
11. Cloudflare: Post-quantum cryptography (PQC). Cloudflare SSL/TLS documentation, https://developers.cloudflare.com/ssl/post-quantum-cryptography/, accessed: 2026-10-22
12. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Miller, G.L. (ed.) STOC. pp. 212–219. ACM (1996). https://doi.org/10.1145/237814.237866

13. Hosoyamada, A.: On post-quantum multi-key security of GCM. Cryptology ePrint Archive, Paper 2026/1718 (2026), https://eprint.iacr.org/2026/1718
14. Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In: Peyrin, T., Galbraith, S.D. (eds.) ASIACRYPT (1). Lecture Notes in Computer Science, vol. 11272, pp. 275–304. Springer (2018). https://doi.org/10.1007/978-3-030-03326-2_10
15. Jaeger, J., Song, F., Tessaro, S.: Quantum key-length extension. In: Nissim, K., Waters, B. (eds.) TCC (1). Lecture Notes in Computer Science, vol. 13042, pp. 209–239. Springer (2021). https://doi.org/10.1007/978-3-030-90459-3_8
16. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symmetric Cryptol. **2016**(1), 71–94 (2016). https://doi.org/10.13154/TOSC.V2016.I1.71-94
17. Kilian, J., Rogaway, P.: How to protect DES against exhaustive key search. In: Koblitz, N. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1109, pp. 252–267. Springer (1996). https://doi.org/10.1007/3-540-68697-5_20
18. Leander, G., May, A.: Grover meets Simon - quantumly attacking the FX-construction. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT (2). Lecture Notes in Computer Science, vol. 10625, pp. 161–178. Springer (2017). https://doi.org/10.1007/978-3-319-70697-9_6
19. Microsoft: BitLocker FAQ. Microsoft Learn, https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/faq, accessed: 2026-10-22
20. Microsoft: Configure BitLocker. Microsoft Learn, https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/configure, accessed: 2026-10-22
21. National Institute of Standards and Technology: Module-lattice-based key-encapsulation mechanism standard. FIPS PUB 203, National Institute of Standards and Technology (Aug 2024). https://doi.org/10.6028/NIST.FIPS.203, https://csrc.nist.gov/pubs/fips/203/final
22. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 9846 (Jul 2026). https://doi.org/10.17487/RFC9846, https://www.rfc-editor.org/info/rfc9846/
23. Shiba, R., Iwata, T.: Multi-key security in the quantum world: Revisiting tweakable Even-Mansour and FX. IACR Trans. Symmetric Cryptol. **2026**(2) (2026). https://doi.org/10.46586/TOSC.ABKSUDSUURH6
24. Shiba, R., Iwata, T.: Post-quantum security of tweakable key-alternating Feistel ciphers in the multi-key setting. In: Galdi, C., Siniscalchi, L. (eds.) SCN. Lecture Notes in Computer Science, vol. 16810, pp. 285–307. Springer (2026). https://doi.org/10.1007/978-3-032-36264-3_15
25. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Goldwasser, S. (ed.) FOCS. pp. 124–134. IEEE Computer Society (1994). https://doi.org/10.1109/SFCS.1994.365700